Model Optimization and Tuning Phase

Date	21 April 2024
Team ID	738184
Project Title	Eye Disease Prediction using Deep Learning
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters

CNN	`model.compile(loss='categorical_crossentropy',`
	`optimizer=RMSprop(learning_rate=0.001),`
	`metrics=['accuracy'])`
	`return model`
	
	Loss Function: loss='categorical_crossentropy' - Determines the error between predicted and actual output for multi-class classification tasks.
	
	Optimizer: optimizer=RMSprop(learning_rate=0.001) - RMSprop optimizer with a learning rate of 0.001, adjusting the step size during training for better convergence.

	Metrics: metrics=['accuracy'] - Evaluation metric to measure the proportion of correctly classified examples out of the total during training and testing.
	
	`detection=model.fit_generator(x_train,steps_per_epoch=46, epochs=100,validation_data=x_test, validation_steps=20)`
	
	After applying 100 epochs we got 76% accuracy so we decided to reduce the epoch
	
	`] detection2=cnn_model.fit_generator(x_train,steps_per_epoch=50, epochs=80,validation_data=x_test, validation_steps=20)`
	
	After applying 80 epochs we got 85% of accuracy so we thought of increasing epoch to increase the accuracy.
	
	`detection1=model.fit_generator(x_train,steps_per_epoch=46, epochs=70,validation_data=x_test, validation_steps=20)`
	
	But after applying 70 epochs we got 82% of accuracy so we decided to go with 80 epoch only.

Xception	`# Compile the model` `model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])`

<table>
<tr><td></td><td>Optimizer (optimizer=Adam(learning_rate=0.001)): Adam optimizer with an initial learning rate of 0.001.

Loss Function (loss='categorical_crossentropy'): Categorical cross-entropy loss function for multi-class classification tasks.

Metrics (metrics=['accuracy']): Accuracy metric to evaluate the model's performance during training and testing.</td></tr>
<tr><td>VGG 19</td><td><pre>from tensorflow.keras.optimizers import RMSprop
optimizer = RMSprop(learning_rate=0.001)
vgg19.compile(loss='categorical_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])</pre>
Optimizer (optimizer = RMSprop(learning_rate=0.001)): RMSprop optimizer with a learning rate of 0.001.

Loss Function (loss='categorical_crossentropy'): Categorical cross-entropy loss function for multi-class classification tasks.

Metrics (metrics=['accuracy']): Accuracy metric to evaluate the model's performance during training and testing.</td></tr>
<tr><td>Inception V3</td><td><pre># Create the final model
model = Model(inputs=base_model.input, outputs=predictions)

Compile the model
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])</pre></td></tr>
</table>

	Optimizer: Adam optimizer is used with a learning rate of 0.001. Adam is an adaptive learning rate optimization algorithm that's widely used for training deep learning models. **Loss Function:** Categorical cross-entropy loss function is used. This is a common choice for multi-class classification problems when the labels are one-hot encoded. It measures the difference between the predicted probability distribution and the true probability distribution of the labels. **Metrics:** Accuracy metric is used to evaluate the model's performance during training and testing. It measures the fraction of correctly classified samples out of the total number of samples.

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
CNN	The decision to select a Convolutional Neural Network (CNN) as the final optimized model for eye disease detection was based on its robust capability to extract relevant features from raw image data and its accuracy around 85%. We have built VGG19, XCEPTION, INCEPTION V3 and CNN out of which CNN had the highest accuracy. To refine its performance and ensure robustness, the CNN model underwent further optimization through techniques such as hyperparameter tuning, data augmentation methods. These enhancements were crucial in enhancing the model's accuracy, reliability, and generalizability for accurately diagnosing eye diseases from image data.